C. AMENDMENTS TO THE CLAIMS

In order to better assist the Examiner with the prosecution of the case, the current pending claims have been included in their entirety for which reconsideration is requested.

1. (Currently Amended) A system for managing a high availability cluster during failover, comprising:

a primary node running a middleware stack for supporting web applications, wherein a plurality of layers of said middleware stack are active, wherein said primary node is assigned a virtual IP address to which requests are directed;

a secondary node running a redundant middleware stack mirroring said plurality of layers of said middleware stack of said primary node, wherein a first selection of said plurality of layers of said redundant middleware stack are active and a second selection of said plurality of layers of said redundant middleware stack are in standby;

a data replication partition shared between said primary node and said secondary node with data accessible to a selection of said plurality of layers of said active middleware stack, wherein said selection of said plurality of layers of said active middleware stack correspond to said second selection of said plurality of layers of said redundant middleware stack in standby; [[and]]

a first power supply dedicated to said primary node;

a second power supply dedicated to said secondary node;

first connection means for connecting said first power supply and said secondary node;

second connection means for connecting said second power supply and said primary node; and

a heartbeat monitor for detecting a failure at said primary node, wherein in response to detecting said failure at said primary node, said heartbeat monitor transfers said virtual IP address from said primary node to said secondary node, said heartbeat monitor sends a shutoff command via said first connection means to turn off power to

said first power supply ~~said primary node~~, said heartbeat monitor remounts said data replication partition for access by said secondary node, said heartbeat monitor activates said second selection of said plurality of layers of said redundant middleware stack requiring access to said data within said data replication partition.

2. (Original) The system according to claim 1 for managing a high availability cluster, wherein said middleware stack and said redundant middleware stack support J2EE compliant web applications.

3. (Original) The system according to claim 1 for managing a high availability cluster, wherein said middleware stack and said redundant middleware stack comprise said plurality of layers comprising a selection from among a load balancer, a web server, a web application server, a messaging server, a monitoring server, and a database control server.

4. (Original) The system according to claim 1 for managing a high availability cluster, wherein said standby selection of said plurality of layers of said redundant middleware comprise at least one from among a messaging server and a database control server.

5. (Original) The system according to claim 1 for managing a high availability cluster, wherein a web application server active on said secondary node before failover points to a virtual IP address of a messaging server and a database control server from said middleware stack of said primary node.

6. (Original) The system according to claim 5 for managing a high availability cluster, wherein responsive to detecting said failure at said primary node, said heartbeat monitor takes over said virtual IP address and configures said second selection of said plurality of layers of said redundant middleware stack with said virtual IP address on startup.

7. (Original) The system according to claim 1 for managing a high availability cluster, wherein a message queue and a relational database are launched in said data replication partition.

8. (Original) The system according to claim 1 for managing a high availability cluster, wherein said data replication partition is a Data Replication Block Device.

9. (Cancelled).

10. (Original) The system according to claim 1 for managing a high availability cluster, further comprising:

a service monitoring daemon for monitoring a status of a plurality of services provided by said middleware stack, wherein responsive to detecting a failure of a particular service from among said plurality of services, said service monitoring daemon restarts another instance of said particular service at said primary node with a particular process identifier.

11. (Original) The system according to claim 1 for managing a high availability cluster, further comprising:

a cross-over cable connected between said primary node and said secondary node via network adapters; and

said heartbeat monitor for sending heartbeat requests via said cross-over cable, wherein in response to not receive a return of a heartbeat request, said heartbeat monitor detects a failure in the non-responsive node.

12. (Original) The system according to claim 1 for managing a high availability cluster, further comprising:

said middleware stack running in said primary node atop a Linux compliant operating system layer.

13. (Original) A system for managing middleware during failover in a high availability cluster, comprising:

a primary node comprising an active load balancing controller assigned to a first virtual IP address, an active HTTP server, an active web application server, an active management queue controller, and an active database controller each monitored by a first heartbeat;

a secondary node comprising a standby redundant load balancing controller, an active redundant HTTP server, an active redundant web application server, a standby redundant management queue controller, and a standby redundant database controller each monitored by a second heartbeat;

a drbd data storage partition shared between said primary node and said secondary node, wherein a message queue and a database implemented by said active management queue controller and said active database controller of said primary node are activated at said drbd data storage partition, and wherein said drbd allows access to said active management queue controller and said active database controller configured at a second virtual IP address; and

a heartbeat monitor for detecting a failure at said primary node, wherein in response to detecting said failure at said primary node, said heartbeat monitor transfers said first virtual IP address from said active load balancing controller to said redundant load balancing controller and activates said redundant load balancing controller, said heartbeat monitor calls STONITH to turn off power to said primary node, said heartbeat monitor activates and assigns said second virtual IP address to said redundant management queue controller and said redundant database controller, and said heartbeat monitor remounts said drbd partition for access by said redundant management queue controller and said redundant database controller.

14. (Cancelled).

15. (Currently Amended) A method for managing a high availability cluster during failover, comprising:

controlling a primary node running a middleware stack for supporting web applications, wherein a plurality of layers of said middleware stack are active, wherein said primary node is assigned a virtual IP address to which requests are directed;

controlling a secondary node running a redundant middleware stack mirroring said plurality of layers of said middleware stack of said primary node, wherein a first selection of said plurality of layers of said redundant middleware stack are active and a second selection of said plurality of layers of said redundant middleware stack are in standby;

managing a data replication partition shared between said primary node and said secondary node with data accessible to a selection of said plurality of layers of said active middleware stack, wherein said selection of said plurality of layers of said active middleware stack correspond to said second selection of said plurality of layers of said redundant middleware stack in standby;

establishing a first connection between a first power supply and said secondary node, wherein said first power supply is dedicated to said primary node;

establishing a second connection between said second power supply and said primary node, wherein said second power supply is dedicated to said secondary node; and

responsive to detecting a failure at said primary node, transferring said virtual IP address from said primary node to said secondary node, sending a shutoff command via said first connection to ~~turning~~ turn off power to said first power supply ~~said primary node~~, remounting said data replication partition for access by said secondary node, activating said second selection of said plurality of layers of said redundant middleware stack which require access to said data within said data replication partition.

16. (Original) The method according to claim 15 for managing a high availability cluster, further comprising:

controlling said middleware stack and said redundant middleware stack to support J2EE compliant web applications.

17. (Original) The method according to claim 15 for managing a high availability cluster, wherein said middleware stack and said redundant middleware stack comprise said plurality of layers comprising a selection from among a load balancer, a web server, a web application server, a messaging server, a monitoring server, and a database control server.

18. (Original) The method according to claim 15 for managing a high availability cluster, wherein said standby selection of said plurality of layers of said redundant middleware comprise at least one from among a messaging server and a database control server.

19. (Original) The method according to claim 15 for managing a high availability cluster, further comprising:

enabling a web application server active on said secondary node before failover points to access a messaging server and a database control server from said middleware stack of said primary node.

20. (Original) The method according to claim 15 for managing a high availability cluster, further comprising:

launching a message queue and a relational database in said data replication partition.

21. (Canceled).

22. (Original) The method according to claim 15 for managing a high availability cluster, further comprising:

monitoring a status of a plurality of services provided by said middleware stack;

responsive to detecting a failure of a particular service from among said plurality of services, restarting another instance of said particular service at said primary node with a particular persistent identifier.

23. (Original) The method according to claim 15 for managing a high availability cluster, further comprising:

sending heartbeat requests between said primary node and said secondary node; and

responsive to not receive a return of a heartbeat request, detecting a failure in the non-responsive node.

24. (Original) The method according to claim 15 for managing a high availability cluster, further comprising:

running said middleware stack in said primary node atop a Linux compliant operating system layer.

25. (Currently Amended) A computer program product, residing on a volatile or non-volatile computer readable medium, for managing a high availability cluster during failover, comprising:

means for controlling a primary node running a middleware stack for supporting web applications, wherein a plurality of layers of said middleware stack are active, wherein said primary node is assigned a virtual IP address to which requests are directed;

means for controlling a secondary node running a redundant middleware stack mirroring said plurality of layers of said middleware stack of said primary node, wherein a first selection of said plurality of layers of said redundant middleware stack are active and a second selection of said plurality of layers of said redundant middleware stack are in standby;

means for managing a data replication partition shared between said primary node and said secondary node with data accessible to a selection of said plurality of layers of said active middleware stack, wherein said selection of said plurality of layers of said active middleware stack correspond to said second selection of said plurality of layers of said redundant middleware stack in standby; and

means for establishing a first connection between a first power supply and said secondary node, wherein said first power supply is dedicated to said primary node;

means for establishing a second connection between said second power supply and said primary node, wherein said second power supply is dedicated to said secondary node; and

means, responsive to detecting a failure at said primary node, for transferring said virtual IP address from said primary node to said secondary node, for sending a shutoff command via said first connection to ~~turning~~ turn off power to said first power supply ~~said primary node~~, remounting said data replication partition for access by said secondary node, activating said second selection of said plurality of layers of said redundant middleware stack which require access to said data in said data replication partition.

26. (Original) The computer program product according to claim 25 for managing a high availability cluster, furthering comprising:

means for controlling said middleware stack and said redundant middleware stack to support J2EE compliant web applications.

27. (Original) The computer program product according to claim 25 for managing a high availability cluster,

means for controlling said middleware stack and said redundant middleware stack comprising a selection from among a load balancer, a web server, a web application server, a messaging server, a monitoring server, and a database control server.

28. (Original) The computer program product according to claim 25 for managing a high availability cluster, further comprising:

means for enabling a web application server active on said secondary node before failover points to access a messaging server and a database control server from said middleware stack of said primary node.

29. (Original) The computer program product according to claim 25 for managing a high availability cluster, further comprising:

means for launching a message queue and a relational database in said data replication partition.

30. (Canceled).

31. (Original) The computer program product according to claim 25 for managing a high availability cluster, further comprising:

means for monitoring a status of a plurality of services provided by said middleware stack;

means, responsive to detecting a failure of a particular service from among said plurality of services, for restarting another instance of said particular service at said primary node with a particular persistent identifier.

32. (Original) The computer program product according to claim 25 for managing a high availability cluster, further comprising:

means for sending heartbeat requests between said primary node and said secondary node; and

means, responsive to not receive a return of a heartbeat request, for detecting a failure in the non-responsive node.

33. (Original) The computer program product according to claim 25 for managing a high availability cluster, further comprising:

means for running said middleware stack in said primary node atop a Linux compliant operating system layer.

34. (Newly Added) A method for managing middleware during failover in a high availability cluster, comprising:

controlling a primary node comprising an active load balancing controller assigned to a first virtual IP address, an active HTTP server, an active web application server, an active management queue controller, and an active database controller each monitored by a first heartbeat;

controlling a secondary node comprising a standby redundant load balancing controller, an active redundant HTTP server, an active redundant web application

server, a standby redundant management queue controller, and a standby redundant database controller each monitored by a second heartbeat;

managing a drbd data storage partition shared between said primary node and said secondary node, wherein a message queue and a database implemented by said active management queue controller and said active database controller of said primary node are activated at said drbd data storage partition, and wherein said drbd allows access to said active management queue controller and said active database controller configured at a second virtual IP address; and

responsive to a heartbeat monitor detecting said failure at said primary node, transferring said first virtual IP address from said active load balancing controller to said redundant load balancing controller and activates said redundant load balancing controller, calling STONITH to turn off power to said primary node, activating and assigning said second virtual IP address to said redundant management queue controller and said redundant database controller, and remounting said drbd partition for access by said redundant management queue controller and said redundant database controller.

35. (Newly Added) A computer program product, residing on a volatile or non-volatile computer readable medium, for managing middleware during failover in a high availability cluster, comprising:

means for controlling a primary node comprising an active load balancing controller assigned to a first virtual IP address, an active HTTP server, an active web application server, an active management queue controller, and an active database controller each monitored by a first heartbeat;

means for controlling a secondary node comprising a standby redundant load balancing controller, an active redundant HTTP server, an active redundant web application server, a standby redundant management queue controller, and a standby redundant database controller each monitored by a second heartbeat;

means for managing a drbd data storage partition shared between said primary node and said secondary node, wherein a message queue and a database implemented by said active management queue controller and said active database controller of said primary node are activated at said drbd data storage partition, and wherein said drbd allows access to said active management queue controller and said active database controller configured at a second virtual IP address; and

means, responsive to a heartbeat monitor detecting said failure at said primary node, for transferring said first virtual IP address from said active load balancing controller to said redundant load balancing controller and activates said redundant load balancing controller, calling STONITH to turn off power to said primary node, activating and assigning said second virtual IP address to said redundant management queue controller and said redundant database controller, and remounting said drbd partition for access by said redundant management queue controller and said redundant database controller.